Downloaded from orbit.dtu.dk on: May 04, 2023

DTU Library

Evaluation of Factorization Methods for Thevenin Equivalent Computations in Real-Time Stability Assessment

Hildebrandt, Christina; Karatas, Bahtiyar Can; Glarbo Muller, Jakob; Johannsson, Hjortur

Published in:
Proceedings of 2018 Power Systems Computation Conference

Link to article, DOI:
10.23919/PSCC.2018.8442893

Publication date:
2018

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Hildebrandt, C., Karatas, B. C., Glarbo Muller, J., & Johannsson, H. (2018). Evaluation of Factorization Methods for Thevenin Equivalent Computations in Real-Time Stability Assessment. In *Proceedings of 2018 Power Systems Computation Conference* (pp. 7 pp.). IEEE. https://doi.org/10.23919/PSCC.2018.8442893

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from orbit.dtu.dk on: Nov 20, 2018

Technical University of Denmark

Evaluation of Factorization Methods for Thévenin Equivalent Computations in Real-Time Stability Assessment

Jørgensen, Christina Hildebrandt Lüthje; Karatas, Bahtiyar Can; Møller, Jakob Glarbo; Jóhannsson, Hjörtur

Published in:
Proceedings of 20th Power System Computation Conference

Publication date:
2018

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Hildebrandt, C. B., Karatas, B. C., Møller, J. G., & Jóhannsson, H. (2018). Evaluation of Factorization Methods for Thévenin Equivalent Computations in Real-Time Stability Assessment. In Proceedings of 20th Power System Computation Conference IEEE.

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Evaluation of Factorization Methods for Thévenin Equivalent Computations in Real-Time Stability Assessment

Christina Hildebrandt, Bahtiyar Can Karatas, Jakob Glarbo Møller and Hjörtur Jóhannsson
Department of Electrical Engineering
Technical University of Denmark
Kgs. Lyngby, Denmark
{chhil, bcakara, jglmo, hjjo}@elektro.dtu.dk

***Abstract*—Thévenin equivalents are used by a range of power system stability indicators, such as the L-index for voltage stability and the aperiodic small signal rotor angle stability indicator. This paper investigates the effect of using different factorization methods for computing coefficients for wide-area Thévenin equivalents. Direct and incomplete factorization methods are compared with respect to runtime, accuracy and amount of fill-in. The paper introduces a proof that the block triangular form of bus admittance matrices will have no non-zero entries in the off-diagonal. KLU factorization is found to perform almost twice as fast as the standard LU factorization with no cost of accuracy. It is, however, shown that the largest computational workload is associated with dense matrix multiplications. An incomplete method reduces the fill-in of coefficient matrices at the cost of accuracy in Thévenin voltages. It is shown, that inaccuracies are amplified as the L-index approaches the stability limit.**

***Index Terms*—Power system analysis computing, Power system stability, Real-time assessment, Thévenin equivalent, Wide-area monitoring**

I. Introduction

Thévenin equivalent methods have been proposed for reliable assessment of several modes of power system instability - including long-term voltage instability and steady-state instability of generators [1], [2]. These two types of instability can be strongly connected, since long-term voltage instability is provoked by trying to supply more power to a load than the maximum power transfer capabilities of the system, while aperiodic generator instability is driven by the maximum power transfer from a generator to the system. These limits describe the bounds of stable steady-state operation of power systems and may be used to identify the set of feasible solutions of power flow problems. Fast and efficient computation of Thévenin equivalents is a necessary condition for the application of such indicators in real-time and on larger scale. With the increasing usage of phasor measurement units (PMU) [3], [4], complex bus voltages and complex branch currents can be obtained at the rate of system frequency and together with information of the system topology, the Thévenin equivalent methods can be applied in real-time.

When using Thévenin equivalent methods one may choose from two general approaches for obtaining the equivalent parameters; methods based on local measurements [5], and methods based on the full state of the system (wide-area measurements) [6]. Both approaches have their advantages and drawbacks. Local assessment is better suited for use in distributed controllers while wide-area methods may be a better choice for central monitoring and control or sensitivity calculations. The scope of this paper is limited to the wide-area methods and it is assumed that a true system state is available. Thereby this paper focus on computational performance only. For studies on the impact of measurement uncertainty the reader is referred to [7].

The factorization method has previously been proven to have an impact on computations for determining Thévenin impedances [8]. However, when determining coefficients for super-position the use of different factorization methods has not previously been evaluated. The super-position principle can be used to determine the contribution of each voltage- or current source on the Thévenin voltage.

In [6] it is shown, how a Schur complement of the bus admittance matrix can be exploited to efficiently obtain Thévenin equivalents seen from all nodes of a meshed system. The Schur complement is in general considered to be dense [9], but as noted in [6], several of the fill-ins are small and seem quite insignificant. This observation is here used to give room for increased degree of sparsity.

This paper investigates the effect of applying different factorization methods when obtaining coefficients for super-position for wide-area Thévenin equivalents. The highest possible degree of sparsity is pursued in order to speed up computations. The investigation will focus on the resulting runtime as well as accuracy. Candidates for factorization methods are the standard LU factorization in MATLAB (UMFPACK) [10], "Clark Kent" LU factorization (KLU) [11] and incomplete LU factorization (ILU) [12]. Ordering scheme candidates are Block Triangular Form and Approximate Minimum Degree (AMD).

The special attributes of the KLU algorithm makes it unsuited for some inverse problems. This paper contributes

with a proof of how KLU can be used for calculations involving Thévenin equivalents. The proof shows that the block triangular form of a bus admittance matrix has no non-zero entries in the off-diagonal blocks. This proof has not been provided in earlier publications.

Section II introduces the voltage stability indicator for loads from [13] and aperiodic small signal rotor angle stability margin for generators from [2] and explains how a Schur complement is used to obtain the Thévenin equivalents as proposed in [6]. Section III introduces the different factorization methods and proves why KLU can be used in computation of Thévenin equivalents. Section IV evaluates the methods by computational time, degree of sparsity and their accuracy of the resulting stability indicators. Furthermore, the error in Thévenin voltages introduced by ILU is investigated. Section V discusses the results and gives some perspectives on further work, while section VI concludes the paper.

II. Background

A. Voltage stability indicator

A local voltage stability margin for loads is defined in [13]. For a node i the local indicator L_i is defined by the node voltage $\overline{V}_i$ and the Thévenin voltage $\overline{V}_{th,i}$ seen from node i

$$L_i = \left| 1 - \frac{\overline{V}_{th,i}}{\overline{V}_i} \right|. \tag{1}$$

For stable situations $L_i \leq 1$ must not be violated for any node i. Hence the global indicator for voltage stability of the entire system is given by

$$L = \max_{i \in cs} \{L_i\}, \tag{2}$$

where cs represent the loads. Voltage instability may be inferred in the case where $L > 1$.

The importance of accuracy when assessing Thévenin equivalents for system stability studies can be easily demonstrated by adding a random vector-error $\overline{\varepsilon}$ to the Thévenin voltage. This gives the following L-index;

$$L_i = \left| 1 - \frac{\overline{V}_{th,i} + \overline{\varepsilon_i}}{\overline{V}_i} \right| = \left| 1 - \frac{\overline{V}_{th,i}}{\overline{V}_i} - \frac{\overline{\varepsilon_i}}{\overline{V}_i} \right|. \tag{3}$$

The closer the system is to voltage instability the lower the magnitude of the node voltage, V_i will be. Therefore, the error will have a larger influence close to the stability boundary. Figure 1 shows the calculation of the L-index represented with phasors. The worst case is an error vector that is orthogonal to $\frac{\overline{V}_{th,i}}{\overline{V}_i}$. This will result in an L-index that indicate the system is more stable, than it actually is.

B. Aperiodic small signal rotor angle stability margin

In [2] a power margin of the injected power to the maximum power injection is defined. This margin describes the distance from a generator's operating point to the stability boundary of aperiodic small signal rotor angle stability. In [14] this margin is reformulated in terms of voltages instead of impedances. A percentage margin to the maximum injectable power is then defined as

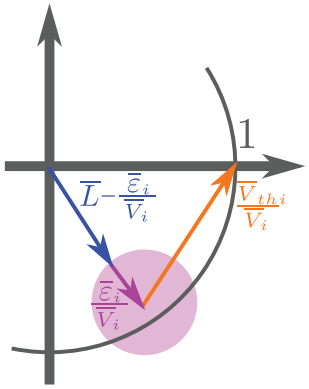


Figure 1. L-index for node i, where an error $\overline{\varepsilon_i}$ affect the resulting stability indicator. The L-index is represented as a phasor in the complex plane. The local voltage stability indicator for the node will be the magnitude of this.

$$\%\Delta P_{inj} = \frac{P_{inj,max} - P_{inj}}{P_{inj}} \cdot 100\% \tag{4}$$

$$= \frac{\cos(\delta + \phi_{th}) + 1}{1 + \frac{V}{V_{th}} \cos \phi_{th}} \cdot 100\%, \tag{5}$$

where the generator is represented as a voltage source $V \angle \delta$ and the remaining grid by its Thévenin equivalent with a voltage source of magnitude V_{th} and an impedance $Z_{th} \angle \phi_{th}$. $\overline{V}_{th}$ is used as the phase angle reference.

If, for any generator, $\%\Delta P_{inj} < 0$, that generator will lose synchronism. This may destabilize the entire system. Therefore, the overall systems stability margin is defined as the minimum $\%\Delta P_{inj}$.

C. Schur complement and Thévenin equivalents

An approach for computing Thévenin equivalents is described in [6], which uses a Schur complement to optimize the calculations. Thévenin equivalents consist of a Thévenin impedance $\overline{Z}_{th}$ and Thévenin voltage $\overline{V}_{th}$. The Thévenin equivalent seen from node i will satisfy

$$\overline{V}_{th,i} = \overline{V}_i - \overline{Z}_{th,i} \overline{I}_i \tag{6}$$

$\overline{V}_i$ is node voltage and $\overline{I}_i$ is current injected at node i.

Nodes in a network can be partitioned in to two sets - current sources (cs) and voltage sources (vs). A floating node may be seen as a current source injecting 0 current. Loads are represented as current sources and generators with automatic voltage regulator (AVR) or internal voltages of manually excited machines as voltage sources. This distinction of nodes is important as it is recalled that the L-index is an indicator of voltage instability of load buses while the aperiodic small signal rotor angle stability margin is associated with the maximum power transfer from generators.

The admittance matrix for the system can then be block-wise partitioned as follows

$$\begin{bmatrix} I_{cs} \\ I_{vs} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{cs} & \mathbf{Y}_{v \to c} \\ \mathbf{Y}_{c \to v} & \mathbf{Y}_{vs} \end{bmatrix} \begin{bmatrix} V_{cs} \\ V_{vs} \end{bmatrix} \tag{7}$$

Eliminating V_{cs} from (7) yields

$$I_{vs} = \mathbf{Y}_{eq} V_{vs} - \mathbf{Q}_{ac} I_{cs} \tag{8}$$

with

$$\mathbf{Y}_{eq} = \mathbf{Y}_{vs} - \mathbf{Y}_{c\to v}\mathbf{Y}_{cs}^{-1}\mathbf{Y}_{v\to c} \tag{9}$$
$$\mathbf{Q}_{ac} = -\mathbf{Y}_{c\to v}\mathbf{Y}_{cs}^{-1} \tag{10}$$

$\mathbf{Y}_{eq}$ is the Schur complement and $\mathbf{Q}_{ac}$ is the accompanying matrix. This reduction of the network is also known as *Kron reduction* [15].

The Thévenin impedances seen from node i are determined as the diagonal of the impedance matrix

$$Z_{th,i} = \begin{cases} \mathbf{Z}_{cs}(i,i) & i \in cs \\ \mathbf{Y}_{eq}(i,i)^{-1} & i \in vs \end{cases} \tag{11}$$

where $\mathbf{Z}_{cs} = \mathbf{Y}_{cs}^{-1}$ [6].

Using the definition for Thévenin voltage given in (6) and the above network equations the Thévenin voltage for the cs and vs nodes respectively are defined as

$$V_{th,cs} = -\mathbf{Z}_{cs}\mathbf{Y}_{v\to c}V_{vs} + (\mathbf{Z}_{cs} - \mathcal{D}(Z_{th,cs}))I_{cs} \tag{12}$$
$$V_{th,vs} = (\mathcal{I} - \mathcal{D}(Z_{th,vs})\mathbf{Y}_{eq})V_{vs} + \mathcal{D}(Z_{th,vs})\mathbf{Q}_{ac}I_{cs} \tag{13}$$

$\mathcal{I}$ is the identity matrix and $\mathcal{D}(Z_{th})$ is the diagonalization of the vector Z_{th} into a diagonal matrix. (12-13) can be written on the form

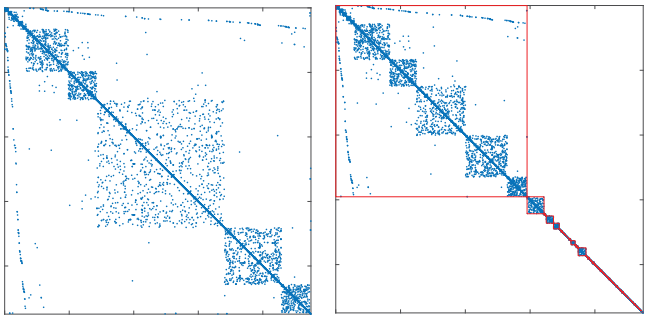
$$\begin{bmatrix} V_{th,cs} \\ V_{th,vs} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_c & \mathbf{K}_{v\to c} \\ \mathbf{Z}_{c\to v} & \mathbf{K}_v \end{bmatrix} \begin{bmatrix} I_{cs} \\ V_{vs} \end{bmatrix} \tag{14}$$

with

$$\mathbf{Z}_c = \mathbf{Z}_{cs} - \mathcal{D}(Z_{th,cs}) \tag{15}$$
$$\mathbf{K}_{v\to c} = -\mathbf{Z}_{cs}\mathbf{Y}_{v\to c} \tag{16}$$
$$\mathbf{Z}_{c\to v} = \mathcal{D}(Z_{th,vs})\mathbf{Q}_{ac} \tag{17}$$
$$\mathbf{K}_v = \mathcal{I} - \mathcal{D}(Z_{th,vs})\mathbf{Y}_{eq} \tag{18}$$

The coefficients $\mathbf{K}_v$ were introduced in [6] and [16] as the grid transformation coefficients (GTC).

Algorithm 1 contains the steps for obtaining the coefficients and Thévenin impedances. The LU-factorization of $\mathbf{Y}_{cs}$ is used to optimize the computations. This approach is used in [6] and [8], since $\mathbf{L}$ and $\mathbf{U}$ are computationally more efficient to invert than the full matrix.



(a) (b)

Figure 2. (a) Sparsity pattern for $\mathbf{Y}_{cs}$ for the test system Polish-Winter03, Table I, used in the Section IV and (b) $\mathbf{Y}_{cs}$ on block triangular form

Algorithm 1 Thévenin equivalents

$\mathbf{L}_{cs}, \mathbf{U}_{cs} \leftarrow$ factorization of $\mathbf{Y}_{cs}$
$\mathbf{U}_{\mathbf{Z}_{cs}} \leftarrow$ solve($\mathbf{L}_{cs}, \mathcal{I}$)
$\mathbf{L}_{\mathbf{Z}_{cs}}^T \leftarrow$ solve($\mathbf{U}_{cs}^T, \mathcal{I}$)
$\mathbf{Z}_{cs} \leftarrow \mathbf{L}_{\mathbf{Z}_{cs}}\mathbf{U}_{\mathbf{Z}_{cs}}$
$Z_{th,cs} \leftarrow \mathcal{D}(\mathbf{Z}_{cs})$
$\mathbf{Q}_{ac} \leftarrow -\mathbf{Y}_{c\to v}\mathbf{Z}_{cs}$
$\mathbf{Y}_{eq} \leftarrow \mathbf{Y}_{vs} + \mathbf{Q}_{ac}\mathbf{Y}_{v\to c}$
$Z_{th,vs} \leftarrow \mathcal{D}(\mathbf{Y}_{eq})^{-1}$
$\mathbf{Z}_c \leftarrow \mathbf{Z}_{cs} - \mathcal{D}(Z_{th,cs})$
$\mathbf{K}_{v\to c} \leftarrow -\mathbf{Z}_{cs}\mathbf{Y}_{v\to c}$
$\mathbf{Z}_{c\to v} \leftarrow \mathcal{D}(Z_{th,vs})\mathbf{Q}_{ac}$
$\mathbf{K}_v \leftarrow \mathcal{I} - \mathcal{D}(Z_{th,vs})\mathbf{Y}_{eq}$
$Z_{th} \leftarrow \begin{bmatrix} Z_{th,cs} \\ Z_{th,vs} \end{bmatrix}$
$\mathbf{K} \leftarrow \begin{bmatrix} \mathbf{Z}_c & \mathbf{K}_{v\to c} \\ \mathbf{Z}_{c\to v} & \mathbf{K}_v \end{bmatrix}$
return Z_{th} and $\mathbf{K}$

$\mathcal{D}(\mathbf{X})$ is a vector containing the diagonal of the matrix $\mathbf{X}$, while $\mathcal{D}(X)$ a diagonal matrix with the vector X along the diagonal.

The Thévenin voltages can be determined by (14) using the coefficients, $\mathbf{K}$, the current injected at cs nodes, $\mathbf{I}_{cs}$, and the voltage at vs nodes, $\mathbf{V}_{vs}$.

III. Factorization Methods

Different factorization methods will be used in Algorithm 1. The different methods investigated are

- UMFPACK with AMD
- KLU using block triangular form
- ILU

UMFPACK with AMD is the standard LU factorization of a sparse matrix in MATLAB. AMD is used prior to factorization, where the matrix is permuted to reduce the computation time and the fill-in in the factorization [17].

KLU is a factorization method optimized for sparse systems [11]. The method is part of the library SuiteSparse [18]. KLU convert the system to block triangular form, where the diagonal of the resulting matrix will contain square matrices with zero-free diagonal and the off-diagonal will contain potentially non-zero blocks.

$$\begin{bmatrix} A_{11} & \cdots & A_{1k} \\ & \ddots & \vdots \\ & & A_{kk} \end{bmatrix} \tag{19}$$

The blocks below the diagonal will be zero. Figure 2 shows $\mathbf{Y}_{cs}$ for the test system Polish-Winter03, Table I and its block triangular form.

The block elements are reordered using AMD and then factorized whereas the off-diagonal elements are kept as is. This gives the following structure for the KLU factorization of a matrix $\mathbf{A}$ [11]

$$\mathbf{PRAQ} = \mathbf{LU} + \mathbf{F} \tag{20}$$

$\mathbf{P}, \mathbf{Q}$ are permutation matrices, $\mathbf{R}$ is a diagonal scaling matrix, $\mathbf{L}, \mathbf{U}$ are the factorization of the diagonal elements and $\mathbf{F}$ represents the entire off-diagonal. KLU use block back substitution to solve a linear system from the factorization in (20). The structure of KLU in the setting of this paper will be treated later.

ILU is an incomplete solver, which will be used to test, how reducing fill-in can speed up the computations and affect the resulting Thévenin voltages. The chosen type of ILU is called the Crout version of ILU (ILUC) [12]. The tolerance for ILU determines when elements are set to 0. Elements will be set to 0, if their value is smaller than the tolerance multiplied by the norm of the column and the tolerance multiplied by the norm of the row. The tolerance in this study is chosen as 10^{-5}. The choice of tolerance is explained in detail in section IV. In addition ILU is set to preserve row sums as it has been found to significantly increase the accuracy of the results without affecting the runtime of the algorithm.

A. KLU of an admittance matrix

The structure of KLU, (20), does not fit in to the setting of Algorithm 1. $\mathbf{Y}_{cs}$ should be split into an $\mathbf{L}$ and $\mathbf{U}$ part, while KLU provides a factorization of the form $\mathbf{LU} + \mathbf{F}$. However, this will not be an issue, since for an admittance matrix with complex admittances $\mathbf{F} = \mathbf{0}$.

In [18] it is stated that the block triangular form of a square matrix, $\mathbf{A}$ with zero-free diagonal corresponds to finding the strongly connected components of a directed graph $G(\mathbf{A})$. An admittance matrix with complex admittances will always have a zero-free diagonal. Therefore, the block triangular form of $\mathbf{Y}_{cs}$ will correspond to finding the strongly connected components of $G(\mathbf{Y}_{cs}) = (V, E)$ with the nodes $V = \{1, \ldots, |cs|\}$ and the edges $E = \{(i, j) \mid \mathbf{Y}_{cs}(i, j) \neq 0\}$.

A strongly connected component is defined as maximal set of nodes such that for any pair of nodes in the set the paths $i \rightsquigarrow j$ and $j \rightsquigarrow i$ exists. This means that there will be a path both from i to j and from j to i in the directed graph.

The non-zero pattern of an admittance matrix is symmetric. This means, that $\mathbf{Y}_{cs}(i, j) \neq 0 \Leftrightarrow \mathbf{Y}_{cs}(j, i) \neq 0$ and in equality $\mathbf{Y}_{cs}(i, j) = 0 \Leftrightarrow \mathbf{Y}_{cs}(j, i) = 0$. Therefore, there are two scenarios for edges between two nodes i and j in the directed graph $G(\mathbf{Y}_{cs})$. Either there will be no edges between the nodes i and j or there will be an edge both from i to j, (i, j), and from j to i, (j, i), see Figure 3. This means that two nodes will either be in the same strongly connected component or they will be completely separated, since the graph only contains bidirectional edges. Hence the block triangular form of $\mathbf{Y}_{cs}$ will consist of the strongly connected components in the diagonal and the entire off-diagonal will be empty, since there is no connection between the components. Edges in the off-diagonal block would stem from asymmetries in the non-zero pattern, where $\mathbf{Y}_{cs}(i, j) \neq 0$ and $\mathbf{Y}_{cs}(j, i) = 0$. Therefore, KLU factorization of $\mathbf{Y}_{cs}$ will always satisfy $\mathbf{F} = \mathbf{0}$.

In the graph for the entire power system there will be a path between any two nodes. The algorithm however only factorize

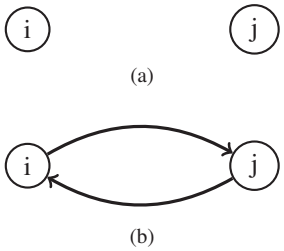


Figure 3. The directed graph $G(\mathbf{Y}_{cs})$ will either have (a) no edges between two nodes i and j or (b) 2 edges (i, j) and (j, i)

$\mathbf{Y}_{cs}$, and the cs nodes need not all be connected, since they can be connected through the vs nodes, which are left out. Figure 4 shows an example of a network divided in to cs and vs nodes. This contain 2 connected components when excluding the vs nodes. Furthermore, it can be noted, that finding the block triangular form will not be an advantage if factorizing the entire admittance matrix. The diagonal will in this case contain only one block element, which is the entire matrix, since all nodes will belong to the same strongly connected component.

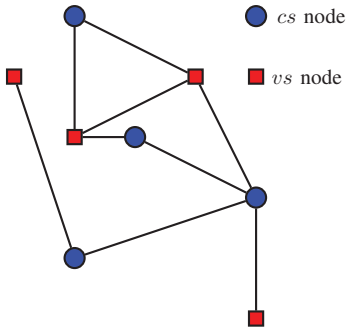


Figure 4. Example of a network divided in to cs and vs nodes

IV. Implementation and test

Algorithm 1 is implemented in MATLAB in order to assess the effect of the different factorization methods. The methods are evaluated with respect to both runtime, accuracy of the results and the number of non-zeros in the resulting coefficients. The runtime is tested on an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.50GHz. The test systems can be seen in Table I.

The Pegase and Polish-Winter systems can be found in MATPOWER, [19], the PTI systems are included in the PSS®E 33.0 examples and Nordic32 can be found in [20].

A. The Tolerance of ILU

Before comparing runtimes, fill-ins and precission of the different factorization methods it is necessary to identify a suitable tolerance for ILU. This tolerance determines the sparsity, accuracy and computation time. Errors in Thévenin voltages obtained with ILU can be stated with results from UMFPACK as reference in the terms of a total vector error (TVE) [21]

$$\text{TVE } (\%) = \sqrt{\frac{(\tilde{X}_r - X_r)^2 + (\tilde{X}_i - X_i)^2}{X_r^2 + X_i^2}} \cdot 100\% \qquad (21)$$

TABLE I
TEST SYSTEMS

Case	no. of buses	no. of vs nodes	non-zeros in Y
Nordic32	46	20	160
Pegase89	89	12	501
Pegase1354	1354	260	4774
PTI-WECC-1648	1648	313	6680
Polish-Winter99	2383	327	8155
Polish-Winter03	2746	374	9344
Pegase2869	2869	510	10805
Polish-Winter07	3012	347	10144
PTI-EECC-7991	7917	1325	32211
Pegase9241	9241	1445	37655

where $\tilde{X}$ is the estimate (ILU) and X is the true value (UMFPACK).

Table II shows the sparsity of ILU for different tolerance levels as well as the runtime of Algorithm 1, runtime for computing Thévenin voltages and the TVE of the Thévenin voltages for the test system Pegase9241. Furthermore, the same information for UMFPACK is included in the table.

TABLE II
PERFORMANCE OF ILU DEPENDING ON CHOICE OF TOLERANCE FOR PEGASE9241

Tolerance	Non-zeros of $\mathbf{K}$	Runtime (s) Algorithm 1	Runtime (ms) V_{th}	Max TVE (%) V_{th}
UMFPACK	42515618	11.95	128.08	-
10^{-3}	9126625	2.14	35.69	4540.43
10^{-4}	29047282	7.13	92.22	14.80
10^{-5}	37618552	17.32	119.92	1.89
10^{-6}	42101927	29.48	126.32	0.27
10^{-7}	42423098	43.80	125.80	0.029
10^{-8}	42498272	47.31	129.70	$2.08{\cdot}10^{-3}$
10^{-9}	42506413	50.67	139.04	$2.28{\cdot}10^{-4}$

Increasing the tolerance increases the sparsity of the coefficient matrix while the time for calculating the Thévenin voltages is reduced along with the accuracy. For a tolerance larger than 10^{-6} the sparsity of $\mathbf{K}$ obtained with ILU is approaching that obtained with UMFPACK and the advantage of using ILU disappears.

At a tolerance of 10^{-4} ILU shows runtimes of Algorithm 1, that are comparable to UMFPACK for all systems. For Pegase9241 the runtime is lower, but the errors in Thévenin voltages at this level of tolerance were shown to be up to 15% TVE. An error in Thévenin voltage of a few percent might be accounted for by defining an appropriate trigger-margin for the stability indicators. Thus, the choice of a suitable tolerance for ILU must satisfy the criteria that there should be an advantage of ILU over UMFPACK in terms of runtime, and the resulting inaccuracy should be a few percent at most. On this basis a tolerance of 10^{-5} is chosen.

With the choice of tolerance in place the accuracy of ILU may be assessed for all the test systems. Figure 5 shows the TVE between ILU and UMFPACK of the Thévenin voltages for all test systems. The mean value of TVE is in general small in all the cases, which means that there are few large errors. The largest vector error is smaller than 2% for all test systems.

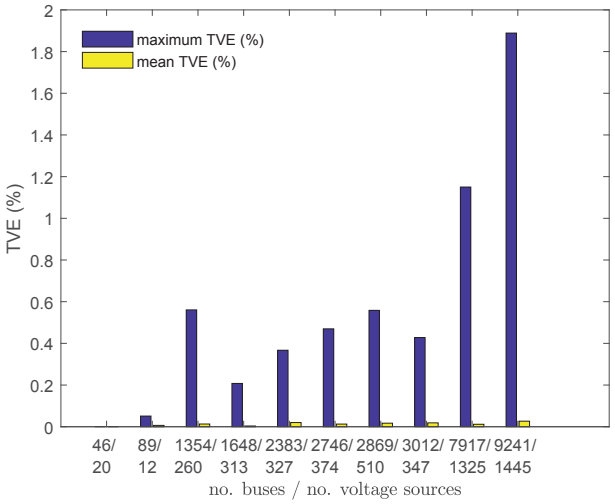


Figure 5. Maximum and mean TVE (%), when comparing Thévenin voltages from ILU to UMFPACK. The tolerance of ILU is set to 10^{-5}.

B. Comparing Factorization Methods

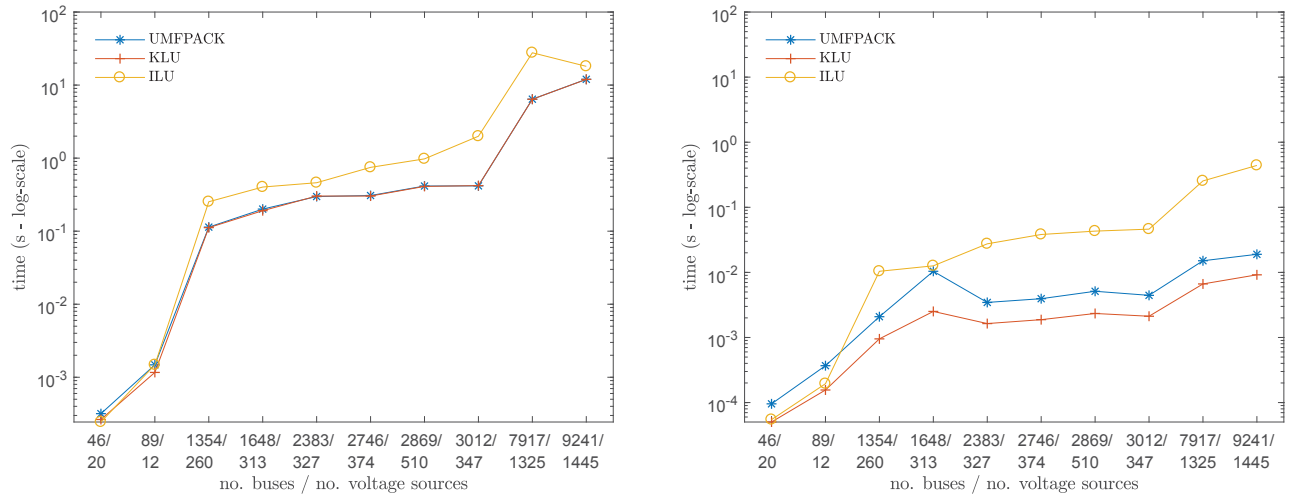
Figure 6a shows the total runtime of Algorithm 1 for each factorization method. The total runtime of UMFPACK and KLU are close with a small favor to KLU. ILU for the chosen tolerance is slower in all but the smallest cases, and the runtime seems to be very dependent on the structure of the network.

Figure 6b shows the runtime for the factorization step of Algorithm 1. KLU consistently performs the factorization almost twice as fast as UMFPACK and is the fastest factorization method in all cases. However, comparing the runtimes reported in Figure 6a and 6b it is evident that the time spent on factorization is negligible compared to the total runtime of Algorithm 1.

Looking at the ability to predict instability issues Table III and IV show that ILU with the chosen settings is close to the results from UMFPACK. The results for KLU show that the accuracy can be considered to be the same as for UMFPACK. In all cases ILU predicted the same node to be the critical node as both UMFPACK and KLU.

For all cases KLU and UMFPACK are close both in computation time and accuracy and in general the largest error of the Thévenin voltages was 10^{-13} when comparing the two. The larger error for ILU is explained by the inaccuracies in Thévenin voltages seen in Figure 5.

One advantage of ILU is the increased sparsity of the coefficient matrices. Table V shows the number of non-zeros of the coefficient matrix for each factorization method. For small systems ILU does not provide additional sparsity however as the systems grows larger it significantly reduces fill-in. The difference with respect to sparsity between KLU and UMFPACK is insignificant.



(a) (b)

Figure 6. For each factorization method versus size of power system (a) shows the total runtime of Algorithm 1 and (b) the runtime of the factorization step.

TABLE III
VOLTAGE STABILITY INDEX FOR LOADS (L-INDEX)

Case	L-index UMFPACK	Difference KLU	Difference ILU
Nordic32	0.183	0	0
Pegase89	0.316	0	$1.25{\cdot}10^{-4}$
Pegase1354	0.212	0	$-2.2{\cdot}10^{-16}$
PTI-WECC-1648	0.283	$-1.7{\cdot}10^{-15}$	$-1.9{\cdot}10^{-4}$
Polish-Winter99	0.066	$-1.2{\cdot}10^{-14}$	$7.39{\cdot}10^{-5}$
Polish-Winter03	0.105	$5.4{\cdot}10^{-15}$	$1.01{\cdot}10^{-4}$
Pegase2869	0.163	$3.9{\cdot}10^{-16}$	$6.04{\cdot}10^{-4}$
Polish-Winter07	0.075	$2.9{\cdot}10^{-15}$	$-6.3{\cdot}10^{-4}$
PTI-EECC-7991	0.311	$6.7{\cdot}10^{-16}$	$-1.1{\cdot}10^{-4}$
Pegase9241	0.176	$-5.3{\cdot}10^{-16}$	$-2.17{\cdot}10^{-3}$

TABLE IV
APERIODIC SMALL SIGNAL ROTOR ANGLE STABILITY MARGIN FOR GENERATORS ($\min \%\Delta P_{inj}$)

Case	$\min \%\Delta P_{inj}$ UMFPACK	Difference KLU	Difference ILU
Nordic32	38.01	0	0
Pegase89	94.33	0	0.004
Pegase1354	81.00	0	0.018
PTI-WECC-1648	32.82	0	0
Polish-Winter99	81.42	$5.0{\cdot}10^{-13}$	0.026
Polish-Winter03	88.79	$4.0{\cdot}10^{-13}$	$9.05{\cdot}10^{-4}$
Pegase2869	63.41	0	0
Polish-Winter07	79.93	$-7.7{\cdot}10^{-13}$	0.009
PTI-EECC-7991	44.54	0	$1.39{\cdot}10^{-4}$
Pegase9241	62.84	0	0

Reduced fill-in can reduce runtime of the Thévenin voltage calculations. In all cases the runtime of computing the Thévenin voltages were smaller for ILU than for the direct methods. If coefficients were to be used for contingency analysis as in [22] the computation time for the contingency assessments could be reduced using ILU. However, the runtime of Algorithm 1 is also increased significantly using ILU.

TABLE V
NON-ZEROS IN COEFFICIENT MATRIX, $\mathbf{K}$, FOR EACH FACTORIZATION METHOD

Case	Non-zeros UMFPACK	Non-zeros KLU	Non-zeros ILU
Nordic32	548	547	549
Pegase89	7668	7663	7664
Pegase1354	1120074	1120078	1025267
PTI-WECC-1648	1706590	1706580	1694777
Polish-Winter99	2825724	2825717	1835089
Polish-Winter03	3027546	3027555	2577363
Pegase2869	2961791	2961793	2790832
Polish-Winter07	4206611	4206582	4095535
PTI-EECC-7991	44852964	44852962	31146092
Pegase9241	42515624	42515659	37618552

V. DISCUSSION OF RESULTS

The use of KLU provides a small speedup of computations compared to UMFPACK at no cost of accuracy in the resulting coefficient matrices. The speedup is achieved through a faster factorization step, but it has been shown that the factorization step is a negligible part of total runtime of computing Thévenin impedances and the coefficient matrix.

The tolerance for ILU was chosen to be 10^{-5} to ensure errors that were smaller than 2% for all test system. It is clear that the error in these cases have a small effect on the resulting stability indicators in Table III and IV. The biggest error might not have occurred on the most critical node or in a direction that did not change the magnitude of the L-index considerably. However, there is no way of guaranteeing, that this will always be the case. The chosen tolerance result in a considerable increase in the runtime of Algorithm 1. For Pegase9241 in Table II the runtime of V_{th} is reduced by 8 ms, while the runtime of the algorithm is increased by 5.4 seconds. This means that for ILU to be an advantage the calculations for

V_{th} should be done more than 675 times before the coefficients needs to be recalculated.

Furthermore, the error in the Thévenin voltages for ILU will have a larger influence near the stability boundary, and here the system topology will also change more rapidly, which will result in recalculations of coefficients to be done more often. Seeing that ILU both introduce an error and give an increased calculation time for coefficients with no major decrease in runtime for Thévenin voltages it severely limits the applicability of ILU.

An alternative method for introducing sparsity could be to compute the full solution and then set elements to zero if they seem to have a small influence on the result. The evaluation of an element's contribution to the result could be done by a norm related tolerance like how ILU sets elements to zero.

Further investigation of calculations of coefficients show that the computational heavy part of the algorithm is determining $\mathbf{Z}_{cs} = \mathbf{Y}_{cs}^{-1}$. Future work with a focus on reducing the runtime of finding the impedance matrix for the cs nodes, would benefit the computations more than a change of factorization method. A sparse implementation of Fox's algorithm [23] was used in [6] to calculate a similar matrix product. Another approach would be to take advantage of backwards solve of KLU. $\mathbf{Z}_{cs}$ could be kept on factorized form, and whenever $\mathbf{Z}_{cs}$ is used in calculations backwards solve would compute the result. In [8] a similar approach is presented, where good performance is obtained, and the method if furthermore optimized by utilizing parallelization.

VI. Conclusion

In this paper different factorization methods for obtaining coefficients for wide-area Thévenin equivalents was evaluated with respect to accuracy, runtime and amount of fill-in.

The results show that the chosen factorization method has little impact on the algorithm for obtaining coefficients, since the factorization step has a short runtime compared to the runtime of the entire algorithm. Looking only at the runtime of the factorization step KLU was the fastest method in all cases. It was furthermore proved, that for an admittance matrix with complex admittances KLU can be used when obtaining coefficients for super-position.

The incomplete factorization method, ILU, was shown to be able to compute stability indicators with an accuracy close to that of UMFPACK and KLU. Using ILU will involve a trade-off between accuracy and sparsity, where sparsity also leads to reduced computation time for both the factorization step and the calculations to determine the Thévenin voltages. For the error of ILU to be in an acceptable range the runtime for computing coefficients is considerably longer than those of UMFPACK and KLU. Furthermore, the increasing influence of errors for a system close to the stability boundary, gives ILU limited applicability.

References

[1] T. Rahman and G. Jasmon, "A new technique for voltage stability analysis in a power system and improved loadflow algorithm for distribution network," in *Proceedings of International Conference on Energy Management and Power Delivery EMPD*, Singapore, 1995, pp. 714–719.

[2] H. Jóhannsson, A. H. Nielsen, and J. Østergaard, "Wide-Area Assessment of Aperiodic Small Signal Rotor Angle Stability in Real-Time," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4545–4557, 2013.

[3] M. Glavic and T. Van Cutsem, "Wide-Area Detection of Voltage Instability From Synchronized Phasor Measurements. Part I: Principle," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1408–1416, 2009.

[4] ——, "Wide-Area Detection of Voltage Instability From Synchronized Phasor Measurements. Part II: Simulation Results," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1417–1425, 2009.

[5] H. Yuan and F. Li, "A comparative study of measurement-based Thevenin equivalents identification methods," in *2014 North American Power Symposium (NAPS)*. IEEE, 2014, pp. 1–6.

[6] J. G. Møller, H. Jóhannsson, and J. Østergaard, "Super-Positioning of Voltage Sources for Fast Assessment of Wide-Area Thévenin Equivalents," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1488–1493, 2017.

[7] A. Perez, J. Moller, H. Johannsson, and J. Ostergaard, "Uncertainty in real-time voltage stability assessment methods based on Thévenin equivalent due to PMU's accuracy," in *Proceedings of the 5th IEEE PES Innovative Smart Grid Technologies (ISGT) Europe Conference*, Istanbul, Turkey, 2014, pp. 1–6.

[8] S. Sommer and H. Jóhannsson, "Real-time thevenin impedance computation," in *Proceedings of IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, 2013, pp. 1–6.

[9] L. Giraud, A. Haidar, and Y. Saad, "Sparse approximations of the Schur complement for parallel algebraic hybrid solvers in 3D," *Numerical Mathematics-theory Methods and Applications*, vol. 3, no. 3, pp. 276–294, 2010.

[10] T. A. Davis, "Algorithm 832: UMFPACK V4.3 - An unsymmetric-pattern multifrontal method," *ACM Transactions on Mathematical Software*, vol. 30, no. 2, pp. 196–199, 2004.

[11] ——, "Algorithm 907 : KLU , A Direct Sparse Solver for Circuit Simulation Problems," *ACM Transactions on Mathematical Software*, vol. 37, no. 3, pp. 1–17, 2010.

[12] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[13] P. Kessel and H. Glavitsch, "Estimating the Voltage Stability of a Power System," *IEEE Trannsactions on Power Delivery*, vol. 1, no. 3, pp. 346–354, 1986.

[14] T. Weckesser, H. Jóhannsson, and J. Østergaard, "Real-Time Remedial Action Against Aperiodic Small Signal Rotor Angle Instability," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 387–396, 2016.

[15] F. Dorfler and F. Bullo, "Kron Reduction of Graphs With Applications to Electrical Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2013.

[16] E. Dmitrova, H. Jóhannsson, and A. H. Nielsen, "Assessment of the impact that individual voltage source has on a generator's stability," in *Proceedings of the 10th International Power and Energy Conference (IPEC)*, Ho Chi Minh City, Vietnam, 2012, pp. 184–189.

[17] P. R. Amestoy, T. A. Davis, and I. S. Duff, "An Approximate Minimum Degree Ordering Algorithm," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 886–905, 1996.

[18] T. A. Davis, *Direct Methods For Sparse Linear Systems*, N. J. Higham, Ed. Gainesville, Florida: SIAM, 2006.

[19] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[20] CIGRÉ TF38.02.08, "Long Term Dynamics Phase II, Final Report," Tech. Rep., 1993.

[21] IEEE Standards Association, "C37.118.1-2011 IEEE Standard for Synchrophasor Measurements for Power Systems." Tech. Rep., 2011.

[22] J. G. Møller, H. Jóhannsson, and J. Østergaard, "Thevenin equivalent method for dynamic contingency assessment," in *Proceedings of IEEE Power & Energy Society's General Meeting*, Denver, CO, USA, 2015, p. 5.

[23] G. C. Fox, S. W. Otto, and A. J. G. Hey, "Matrix algorithms on a hypercube I: Matrix multiplication," *Parallel Computing*, vol. 4, no. 1, 1987.